**Test Case 1:** Read the TMP117 temperature sensor value on Raspberry Pi 5 platform by writing a real-time embedded program in C++ using blocking I/O

AC1: Connect the TMP117 to Raspberry Pi 5 using its GPIO pin layout and see TMP117 is receiving the power signal.

AC2: Verify that your program can read the Data Ready interrupt signal sent by TMP117 sensor on GPIO pin.

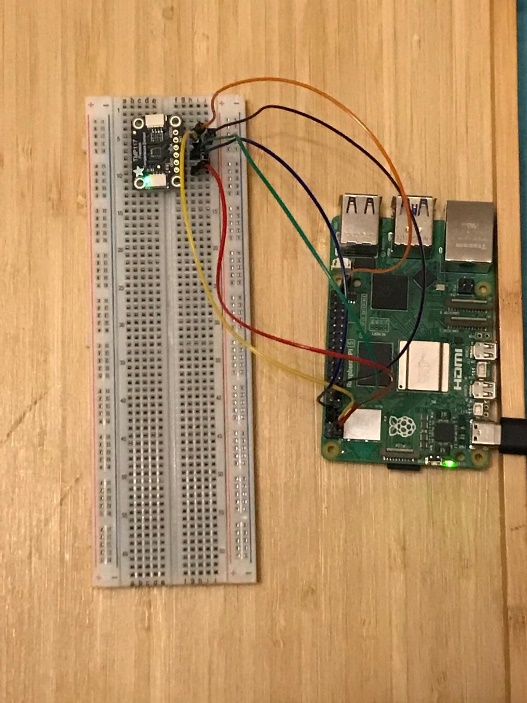
AC3: Verify Raspberry Pi can detect TMP117 on I2C

AC4: Verify your program can read the data from I2C device file for TMP117

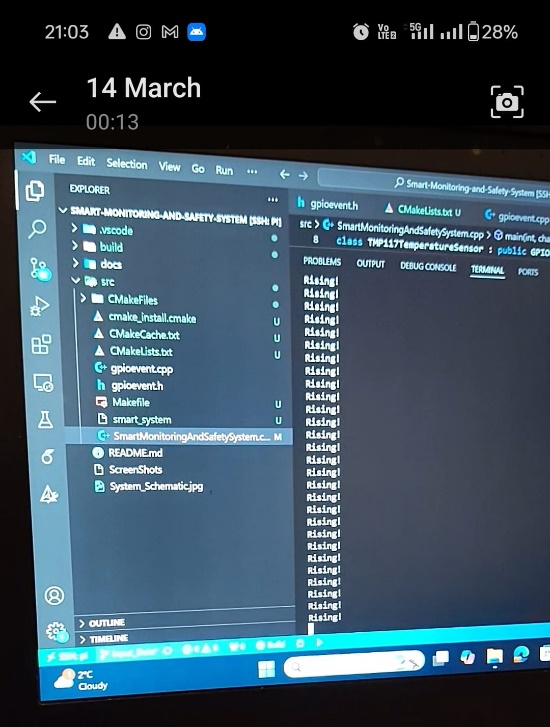
AC5: Verify that the data read from TMP117 sensor is human readable.

Test Case Result: This test case failed multiple times. This document is only showing the final successful result for reference purpose.

AC1:



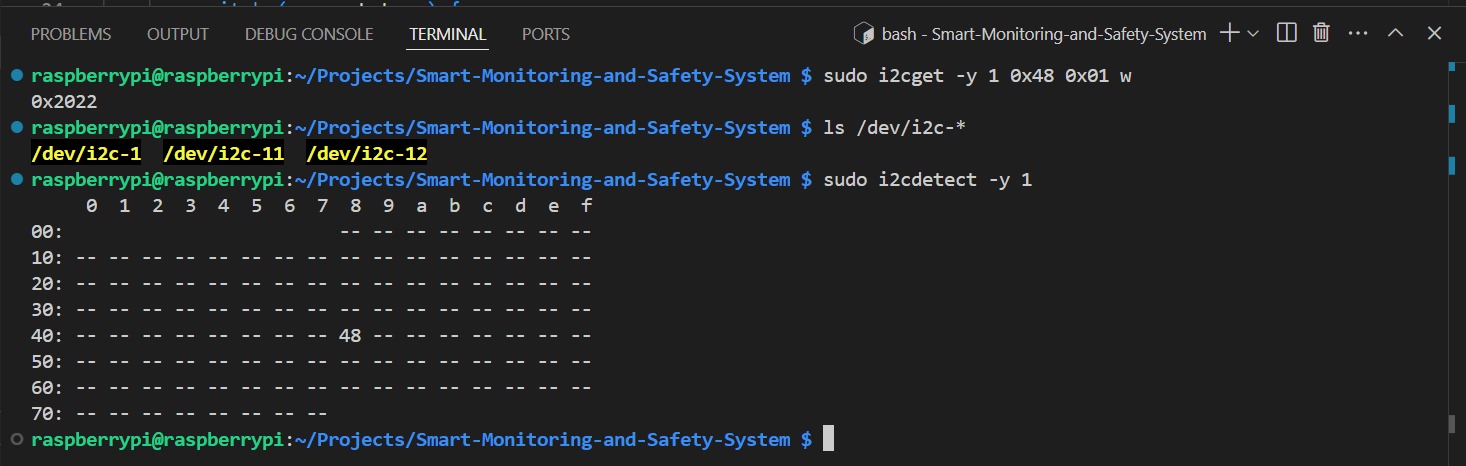
AC2:



AC3:

$ sudo i2cget -y 1 0x48 0x01 w

|  |  |
| --- | --- |
| **Part** | **Meaning** |
| sudo | Runs the command with **root privileges**, required for accessing I2C bus device files. |
| i2cget | Linux **I2C utility command** to **read data** from an I2C device. |
| -y | **Disable interactive confirmation prompt** (automates the command). |
| 1 | Specifies **I2C bus number** → /dev/i2c-1 (Raspberry Pi’s main I2C bus). |
| 0x48 | **7-bit I2C slave address** of the target device → TMP117 default address = 0x48. |
| 0x01 | **Register address to read from** → 0x01 is the TMP117 **Configuration Register**. |
| w | **Word read** → Read **16-bit (2 bytes)** data in one operation (big-endian). |



AC4 & AC5:

